

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

AI
1. (currently amended) A method for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the method steps

comprising:

- a) moving a subset of records within the database table;
- b) flagging each moved record as a reorganization record;
- c) creating a reorganization pointer record for each moved record at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record; and
- d) establishing scanner process constraints based on whether a scanner process is commenced prior to or after the moving, wherein the scanner process can correctly retrieve records from the database table during the reorganization of the database table, including before and after movement of records in the reorganization.

2. (original) The method of claim 1 wherein the establishing further comprises:

- d1) identifying table and index scanner processes commencing prior to the completion of the moving as old table scanners wherein old table scanners are constrained to access records in the database table by following reorganization pointers to access reorganization records and wherein old table scanners are constrained to ignore reorganization records when accessing records sequentially in a scan of the database table;

AI
and

d2) identifying table and index scanner processes commencing after the completion of the moving as new table scanners, wherein new table scanners are constrained to ignore reorganization pointers in accessing records in the database table.

3. (original) The method of claim 2 further comprising the steps of:

e) deleting the reorganization pointer records and removing the reorganization record flag from the records flagged after each query associated with an old table scanner is complete; and

f) repeating steps a-d after the completion of each query associated with a new scanner until the records in the database table meet a reorganization criteria.

4. (original) The method of claim 3 in which step f) alternates between moving records from a one of successively defined portions of the table and moving records in a predefined order into a one of the successively defined vacated portions of the table.

5. (original) The method of claim 4 in which the predefined order for moving records is defined by a clustering index associated with the table.

6. (original) The method of claim 4 in which the table further comprises overflow pointer records pointing to overflow records and in which the method steps further comprise steps defining each overflow pointer record in a one of the successively defined portions of the

AI
table to be a reorganization overflow pointer record, and flagging the associated overflow record to be a reorganization record.

7. (original) The method of claim 4 in which the method steps further comprise the step of preventing the allocation of space for new or updated records in each respective one of the successively defined portions of the table when moving records from the one of successively defined portions of the table.

8. (original) The method of claim 3 wherein the deleting further comprises requesting and waiting for the exclusive availability of a latch which is shareable by each old table scanner process and which each old table scanner process obtains before commencing scanning and releases after the query associated with the old table scanner process has completed processing.

9. (original) The method of claim 8 in which the latch is shareable by each index scanner process and each index scanner process obtains the latch before commencing scanning and releases the latch after the query associated with the index scanner process has completed processing.

10. (original) The method of claim 3 in which the method steps further comprise steps to maintain the identity and status of moved records in a record identifier mapping table.

11. (original) The method of claim 10 in which the record identifier mapping table

A |
comprises a mapping record corresponding to each of the moved records, each mapping record comprising an old record identifier for the initial position of the moved record, a new record identifier for the moved position of the moved record, and a status flag for the moved record.

12. (original) The method of claim 11 in which the status flag for the moved record comprises flags to indicate the completion of the move of the record, the completion of the clean up for the record, and whether the record has been deleted.

13. (original) The method of claim 10 in which the record identifier mapping table comprises a hash table implemented having a hash on the new record identifier, and a hash table containing a reverse record identifier mapping.

14. (original) The method of claim 3 in which the method steps further comprise logging steps for retaining data values relating to:

- 1) the start of step a),
- 2) the move of a normal record,
- 3) the move of a pointer record where, when moving records from a one of successively defined portions of the table, the associated overflow record is not in the vacate range,
- 4) the move of a pointer record where, when moving records from a one of successively defined portions of the table, the associated overflow record is in the vacate range,

- AI
- 5) the move of an overflow record during a vacate phase,
 - 6) the successful move of a record,
 - 7) the successful clean up of a row that has been moved, and
 - 8) the completion of the online reorganization,

wherein the table may be rolled back without the loss of data due to the reorganization of the table.

15. (original) The method of claim 3 in which the method steps further comprise the step of imposing a key value lock where an index scanner is accessing an ordered list of key values in an index associated with the table, the key value lock preventing index values from being updated while the index scanner accesses the ordered list of key values.

16. (original) The method of claim 3 in which the method steps further comprise the step of synchronising between scanners by imposing a share lock on records subject to the moving and records subject to the deleting.

17. (original) The method of claim 3 in which the method steps further comprise an idle step in which neither the moving nor the deleting is performed.

18. (currently amended) A method for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the method comprising the steps of:

- a) a vacate move step that relocates a subset of records in a defined set of pages to

available space within the database table using temporary pointers;

A1

b) a vacate clean up step that removes the temporary pointers of the vacate move step;

c) a fill move step that moves selected records to the defined set of pages using temporary pointers; and

d) a fill clean up step that removes the temporary pointers of the fill move step, wherein each clean up step is synchronised to commence at the completion of a move step and to commence only when all queries launching scanner processes commenced before the completion of a move step have completed and each move step is synchronised to commence at the completion of a clean up step and to commence only when all queries launching scanner processes after the completion of a previous move step have completed.

19. (original) The method of claim 18 wherein step a) comprises:

i) moving each data record in a defined set of pages to a new location in a page in the database table outside the defined set of pages;

ii) flagging each moved non-overflow data record as a reorganization record;

iii) creating a reorganization pointer record for each moved non-overflow data record at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record;

iv) for each overflow pointer record located in the defined set of pages and associated with a moved overflow data record, creating a reorganization pointer record for the moved overflow data record at the location of the said reorganization pointer record and flagging the moved overflow record as a reorganization record; and

AI

v) flagging each overflow pointer record pointing to an overflow record outside the defined set of pages to be a reorganization pointer record, and flagging each of said overflow records as a reorganization record.

20. (original) The method of claim 19 wherein step b) comprises:

- i) deleting reorganization pointers in the defined set of pages; and
- ii) removing the reorganization record flag from each reorganization record in the database table.

21. (original) The method of claim 20 wherein step c) comprises:

- i) moving selected records to the defined set of pages such that the relative ordering of the selected records matches the predefined ordering constraint;
- ii) flagging each moved record as a reorganization record;
- iii) creating a reorganization pointer record for each moved non-overflow record at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record;
- iv) creating a reorganization pointer record for each moved overflow record at the location of the overflow pointer record for the moved record, the reorganization pointer record pointing to the new location of the moved record.

22. (original) The method of claim 21 wherein step d) comprises:

- i) deleting reorganization pointers in the defined set of pages; and
- ii) removing the reorganization record flag from each reorganization record in

the database table.

A1

23. (original) The method of claim 22, wherein table scanner processes commencing during each move step are identified as old table scanners, wherein old table scanners are constrained to access records in the database table by following reorganization pointers to access reorganization records and are further constrained to access reorganization records sequentially by ignoring reorganization records and table scanner processes commencing after the completion of a move step are identified as new table scanners, wherein new table scanners ignore reorganization pointers in accessing records in the database table.

24. (original) The method of claim 22 in which each clean up synchronisation step comprises requesting and waiting for the exclusive availability of a latch which is shareable by each old table scanner process and which each old table scanner process obtains before commencing scanning.

25. (original) The method of claim 22 in which the method further comprises the step of preventing the allocation of space for new or updated records in each respective one of a sequentially advancing defined set of pages during the vacate move step for the said one of the sequentially advancing defined set of pages.

26. (original) The method of claim 22 in which an identity and status of moved records is maintained in a record identifier mapping table.

A1
27. (original) The method of claim 26 in which the record identifier mapping table comprises a mapping record corresponding to each of the moved records, each mapping record comprising an old record identifier for the initial position of the moved record, a new record identifier for the moved position of the moved record, and a status flag for the moved record.

28. (original) The method of claim 27 in which the record identifier mapping table comprises a hash bucket comprising a hash on the new record identifier, and a hash table containing a reverse record identifier mapping.

29. (original) The method of claim 22 in which the database table has an associated index and in which each fill step selects records by scanning the associated index of the database table.

30. (currently amended) A method for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the method comprising the following steps:

a) a vacate move step to move data records from move pages in the table to available space in the database table; and

b) a fill move step to move data records into move pages in the table,
wherein the database table can be accessed, scanned, and updated during the vacate move step and fill move step of the reorganization.

31. (original) The method of claim 30 wherein each move step comprises the step of defining temporary pointers from the original position of each moved record to the moved position of the

AI
moved record.

32. (original) The method of claim 31 further comprising the step of

c) defining selected scanners to be constrained to follow the temporary pointers while accessing records so as to maintain data ordering for the selected scanners wherein the move steps are synchronised with query processes launching scanners so as to maintain the temporary pointers for use by said scanners launched by query processes.

33. (original) The method of claim 32 in which the database table comprises overflow pointer records and the original position of a moved record from which a temporary pointer points.

34. (currently amended) A computer readable medium containing program instructions for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the program instructions comprising the steps of:

- a) moving a subset of records within the database table;
- b) flagging each moved record in as a reorganization record;
- c) creating a reorganization pointer record for each moved record in at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record; and
- d) establishing scanner process constraints based on whether a scanner process is commenced prior to or after the moving, wherein the scanner process can correctly retrieve records from the database table during the reorganization of the database table, including before and after movement of records in the reorganization.

A1
35. (original) The computer readable medium of claim 34 wherein step d) further comprises:

d1) identifying table and index scanner processes commencing prior to the completion of the moving as old table scanners wherein old table scanners are constrained to access records in the database table by following reorganization pointers to access reorganization records and wherein old table scanners are constrained to ignore reorganization records when accessing records sequentially in a scan of the database table; and

d2) identifying table and index scanner processes commencing after the completion of the moving as new table scanners, wherein new table scanners are constrained to ignore reorganization pointers in accessing records in the database table.

36. (original) The computer readable medium of claim 35 further comprising the steps of:

e) deleting the reorganization pointer records and removing the reorganization record flag from the records flagged after each query associated with an old table scanner is complete; and

f) repeating steps a-d after the completion of each query associated with a new scanner until the records in the database table meet a reorganization criteria.

37. (original) The computer readable medium of claim 36 in which step f) alternates between moving records from a one of successively defined portions of the table and moving records in a predefined order into a one of the successively defined vacated portions of the table.

AI

38. (currently amended) A computer readable medium containing program instructions for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the program instructions comprising the steps of:

a) a vacate move step that relocates a subset of records in a defined set of pages to available space within the database table using temporary pointers;

b) a vacate clean up step that removes the temporary pointers of the vacate move step;

c) a fill move step that moves selected records to the defined set of pages using temporary pointers; and

d) a fill clean up step that removes the temporary pointers of the fill move step;

wherein each clean up step is synchronised to commence at the completion of a move step and to commence only when all queries launching scanner processes commenced before the completion of a move step have completed and each move step is synchronised to commence at the completion of a clean up step and to commence only when all queries launching scanner processes after the completion of a previous move step have completed.

39. (original) The computer readable medium of claim 38 wherein step a) comprises:

ii) moving each data record in a defined set of pages to a new location in a page in the database table outside the defined set of pages;

ii) flagging each moved non-overflow data record as a reorganization record;

iii) creating a reorganization pointer record for each moved non-overflow data record at the initial location of the moved record, the reorganization pointer record pointing

AI
to the new location of the moved record;

iv) for each overflow pointer record located in the defined set of pages and associated with a moved overflow data record, creating a reorganization pointer record for the moved overflow data record at the location of the said reorganization pointer record and flagging the moved overflow record as a reorganization record; and

v) flagging each overflow pointer record pointing to an overflow record outside the defined set of pages to be a reorganization pointer record, and flagging each of said overflow records as a reorganization record.

40. (original) The computer readable medium of claim 39 wherein step b) comprises:

- i) deleting reorganization pointers in the defined set of pages; and
- ii) removing the reorganization record flag from each reorganization record in the database table.

41. (original) The computer readable medium of claim 40 wherein step c) comprises:

- i) moving selected records to the defined set of pages such that the relative ordering of the selected records matches the predefined ordering constraint;
- ii) flagging each moved record as a reorganization record;
- iii) creating a reorganization pointer record for each moved non-overflow record at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record; and
- iv) creating a reorganization pointer record for each moved overflow record at the location of the overflow pointer record for the moved record, the reorganization pointer record

AI pointing to the new location of the moved record.

42. (original) The computer readable medium of claim 41 wherein step d) comprises:

- i) deleting reorganization pointers in the defined set of pages; and
- ii) removing the reorganization record flag from each reorganization record in the database table.

43. (original) The computer readable medium of claim 42, wherein table scanner processes commencing during each move step are identified as old table scanners, wherein old table scanners are constrained to access records in the database table by following reorganization pointers to access reorganization records and are further constrained to access reorganization records sequentially by ignoring reorganization records and table scanner processes commencing after the completion of a move step are identified as new table scanners, wherein new table scanners ignore reorganization pointers in accessing records in the database table.

44. (currently amended) A computer readable medium including program instructions implementing steps for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the method steps comprising the following steps:

- a) a vacate move step to move data records from move pages in the table to available space in the database table; and
- b) a fill move step to move data records into move pages in the table, wherein the database table can be accessed, scanned, and updated during the vacate move

A1

step and fill move step of the reorganization.

45. (original) The computer readable medium of claim 44 wherein each move step comprises the step of defining temporary pointers from the original position of each moved record to the moved position of the moved record.

46. (original) The computer readable medium of claim 45 further comprising the step of

c) defining selected scanners to be constrained to follow the temporary pointers while accessing records so as to maintain data ordering for the selected scanners wherein the move steps are synchronised with query processes launching scanners so as to maintain the temporary pointers for use by said scanners launched by query processes.

47. (currently amended) A computer system for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the computer system comprising

means for moving a subset of records within the database table;

means for flagging each record moved by the means for moving a subset of records as a reorganization record;

means for creating a reorganization pointer record for each record moved by the means for moving a subset of records at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record; and

means for establishing scanner process constraints based on whether a scanner process is commenced prior to or after moving the subset of records, wherein the scanner process can

A1

correctly retrieve records from the database table during the reorganization of the database table,
including before and after movement of records in the reorganization.

48. (original) The computer system of claim 47 wherein the means for establishing scanner process constraints further comprises:

means for identifying table and index scanner processes commencing prior to moving the subset of records as old table scanners wherein old table scanners are constrained to access records in the database table by following reorganization pointers to access reorganization records and wherein old table scanners are constrained to ignore reorganization records when accessing records sequentially in a scan of the database table; and

means for identifying table and index scanner processes commencing after the completion of moving the subset of records as new table scanners, wherein new table scanners are constrained to ignore reorganization pointers in accessing records in the database table.

49. (original) The computer system of claim 48 further comprising:

means for deleting the reorganization pointer records and removing the reorganization record flag from the records by the means for flagging after each query associated with an old table scanner is complete; and

means for moving another subset of records after the completion of each query associated with a new scanner until the records in the database table meet a reorganization criteria.

A |

50. (original) The computer system of claim 49 in which the means for moving another subset of records alternates between moving records from a one of successively defined portions of the table and moving records in a predefined order into a one of the successively defined vacated portions of the table.

51. (currently amended) A computer system for reorganizing a database table online , allowing the database table to be scanned, accessed and updated during the reorganization, the computer system comprising:

means for performing a vacate move step that relocates a subset of records in a defined set of pages to available space within the database table using temporary pointers;

means for performing a vacate clean up step that removes the temporary pointers of the vacate move step;

means for performing a fill move step that moves selected records to the defined set of pages using temporary pointers; and

means for performing a fill clean up step that removes the temporary pointers of the fill move step, wherein each clean up step is synchronised to commence at the completion of a move step and to commence only when all queries launching scanner processes commenced before the completion of a move step have completed and each move step is synchronised to commence at the completion of a clean up step and to commence only when all queries launching scanner processes after the completion of a previous move step have completed.

52. (original) The computer system of claim 51 wherein the means for performing the vacate move step comprises:

AI
means for moving each data record in a defined set of pages to a new location in a page in the database table outside the defined set of pages;

means for flagging each moved non-overflow data record as a reorganization record;

means for creating a reorganization pointer record for each moved non-overflow data record at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record;

means for each overflow pointer record located in the defined set of pages and associated with a moved overflow data record, creating a reorganization pointer record for the moved overflow data record at the location of the said reorganization pointer record and flagging the moved overflow record as a reorganization record; and

means for flagging each overflow pointer record pointing to an overflow record outside the defined set of pages to be a reorganization pointer record, and flagging each of said overflow records as a reorganization record.

53. (original) The computer system of claim 52 wherein the means for performing the vacate clean up step comprises:

means for deleting reorganization pointers in the defined set of pages; and

means for removing the reorganization record flag from each reorganization record in the database table.

54. (original) The computer system of claim 53 wherein the means for performing the fill move step comprises:

means for moving selected records to the defined set of pages such that the relative

ordering of the selected records matches the predefined ordering constraint;

means for flagging each moved record as a reorganization record;

means for creating a reorganization pointer record for each moved non-overflow record at the initial location of the moved record, the reorganization pointer record pointing to the new location of the moved record; and

means creating a reorganization pointer record for each moved overflow record at the location of the overflow pointer record for the moved record, the reorganization pointer record pointing to the new location of the moved record.

55. (original) The computer system of claim 54 wherein the means for performing the fill clean up step comprises:

means for deleting reorganization pointers in the defined set of pages; and

means for removing the reorganization record flag from each reorganization record in the database table.

56. (original) The computer system of claim 55 further comprising:

means for identifying table scanner processes commencing during the performance of each move step as old table scanners, wherein old table scanners are constrained to access records in the database table by following reorganization pointers to access reorganization records and are further constrained to access reorganization records sequentially by ignoring reorganization records; and

means for identifying table scanner processes commencing after the completion of the performance of a move step as new table scanners, wherein new table scanners ignore

reorganization pointers in accessing records in the database table.

A1

57. (currently amended) A computer system for reorganizing a database table online, allowing the database table to be scanned, accessed and updated during the reorganization, the computer system comprising:

a) means for carrying out a vacate move step to move data records from move pages in the table to available space in the database table; and

b) means for carrying out a fill move step to move data records into the move pages in the table,

wherein the database table can be accessed, scanned, and updated during the vacate move step and fill move step of the reorganization.

58. (original) The computer system of claim 57 further comprising:

means for defining temporary pointers from an original position of each moved record to the moved position of the moved record;

means for defining selected scanners to be constrained to follow the temporary pointers while accessing records so as to maintain data ordering for the selected scanners;
and

means for synchronising the move steps with query processes launching scanners so as to maintain the temporary pointers for use by said scanners launched by query processes.

59. (original) The computer system of claim 58 in which the database table comprises overflow pointer records and the computer system further comprises means for setting the

A1 original position of a moved record to be the position of an overflow pointer record, from which a temporary pointer points, where the moved data record is originally pointed to by said overflow pointer record.
